

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the application of:)
Byung Joon Lee et al.,) Examiner:
Application No.:)
Filed:)
For: XML-BASED NETWORK MANAGEMENT)
SYSTEM AND METHOD FOR)
CONFIGURATION MANAGEMENT OF)
HETEROGENEOUS NETWORK DEVICES) Art Group:

PRIORITY DOCUMENT SUBMITTAL

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

Submitted herewith is a document upon which Applicant respectfully requests a convention priority for the above-captioned application, namely a certified copy of Korean Patent Application No. 2002-78703 filed December 11, 2002.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: July 3, 2003

By:

Eric S. Hyman,

Reg. No. 30,139

CERTIFICATE OF MAILING:

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class Mail, with sufficient postage, in an envelope addressed to: Mail Stop NEW APPLICATIONS, FEE, Commissioner For Patents P.O. Box 1450, Alexandria, VA 22313-1450

Linda Marie D'Elia

July , 2003

ESH/lmd

**KOREAN INTELLECTUAL
PROPERTY OFFICE**

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

Application Number:: Korean Patent Application 2002-0078703

Date of Application:: 11 December 2002

Applicant(s): : Electronics and Telecommunications Research Institute

30 December 2002

COMMISSIONER

[Bibliography]

[Document Name] Patent Application
[Classification] Patent
[Receiver] Commissioner
[Reference No.] 0010
[Filing Date] 11 December2002
[IPC] H04L

[Title] XML based network management system and method for configuration management of heterogeneous network devices

[Applicant] Electronics & Telecommunications Research Institute
[Name] 3-1998-007763-8
[Applicant code]

[Attorney] Youngpil Lee
[Name] 9-1998-000334-6
[Attorney code]
[General Power of Attorney] 2001-038378-6
Registration No.]

[Attorney] Haeyoung Lee
[Name] 9-1999-000227-4
[Attorney code]
[General Power of Attorney] 2001-038396-8
Registration No.]

[Inventor] LEE, Byung Joon
[Name]
[Resident] 720303-1057415
Registration No.]
[Zip Code] 302-823
[Address] 3 F.,158-2 Doma 2-dong, Seo-gu, Daejeon-city, Rep. of Korea
[Nationality] Republic of Korea

[Inventor] CHOI, Tae Sang
[Name]
[Resident] 620720-1690217
Registration No.]
[Zip Code] 305-761
[Address] 106-707 Expo Apt. 464-1 Jeonmin-dong, Yusong-gu
Daejeon-city, Rep. of Korea
[Nationality] Republic of Korea

[Inventor]
 [Name] JEONG, Tae Soo
 [Resident]
 Registration No.] 570923-1691716
 [Zip Code] 305-755
 [Address] 132-402 Hanbit Apt. Eoeun-dong, Yusong-gu, Daejeon-city
 Rep. of Korea
 Republic of Korea
 [Nationality]

[Type of publication] Disclosed in a written form at academic organization
 [Date of publication] 25 July 2002

[Type of publication] Disclosed in a written form at academic organization
 [Date of publication] 25 September 2002

[Request for Examination] Requested

[Purpose] We file as above according to Art. 42 of the Patent Law
 request the examination as above according to Art. 60
 of the Patent Law.
 Attorney Youngpil Lee
 Attorney Haeyoung Lee

[Fee]
 [Basic page] 20 Sheet(s) 29,000 won
 [Additional page] 10 Sheet(S) 10,000 won
 [Priority claiming fee] 0 Case(S) 0 won
 [Examination fee] 19 Claim(s) 717,000 won
 [Total] 756,000 won
 [Reason for Reduction] Government Invested Research Institution
 [Fee after Reduction] 378,000 won

[Transfer of Technology]
 [Assignment of Technology] Allowable
 [Licensing] Allowable
 [Technology Training] Allowable

[Enclosures]

1. Abstract and Specification (and Drawings) 1 copy
2. Certificate for the application of rules stipulating exception of being well known (exception of losing novelty, exceptional standing at filing) 2 copy

대한민국 특허청
KOREAN INTELLECTUAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2002-0078703
Application Number PATENT-2002-0078703

출원년월일 : 2002년 12월 11일
Date of Application DEC 11, 2002

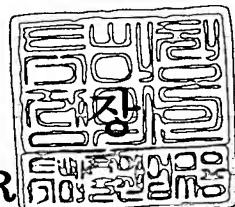
출원인 : 한국전자통신연구원
Applicant(s) Electronics and Telecommunications Research Institute



2002년 12월 30일

특허청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0010
【제출일자】	2002.12.11
【국제특허분류】	H04L
【발명의 명칭】	이기종 네트워크 장비의 구성 관리를 위한 엑스엠엘 기반 망 관리 시스템 및 방법
【발명의 영문명칭】	XML based network management system and method for configuration management of heterogeneous network devices
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2001-038378-6
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2001-038396-8
【발명자】	
【성명의 국문표기】	이병준
【성명의 영문표기】	LEE, Byung Joon
【주민등록번호】	720303-1057415
【우편번호】	302-823
【주소】	대전광역시 서구 도마2동 158-2 3층
【국적】	KR
【발명자】	
【성명의 국문표기】	최태상
【성명의 영문표기】	CHOI, Tae Sang
【주민등록번호】	620720-1690217

【우편번호】	305-761
【주소】	대전광역시 유성구 전민동 464-1 엑스포아파트 106-707
【국적】	KR
【발명자】	
【성명의 국문표기】	정태수
【성명의 영문표기】	JEONG, Tae Soo
【주민등록번호】	570923-1691716
【우편번호】	305-755
【주소】	대전광역시 유성구 어은동 한빛아파트 132동 402호
【국적】	KR
【공개형태】	학술단체 서면발표
【공개일자】	2002.07.25
【공개형태】	학술단체 서면발표
【공개일자】	2002.09.25
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이영필 (인) 대리인 이해영 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	10 면 10,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	19 항 717,000 원
【합계】	756,000 원
【감면사유】	정부출연연구기관
【감면후 수수료】	378,000 원
【기술이전】	
【기술양도】	희망
【실시권 허여】	희망
【기술지도】	희망
【첨부서류】	1. 요약서·명세서(도면)_1통 2. 공지예외적용대상(신규성상 실의예외, 출원시의특례)규정을 적용받기 위한 증명서류_2통

【요약서】**【요약】**

본 발명은 망 관리 시스템의 유지 보수성을 향상시킬 수 있는 엑스엠엘(Extensible Markup Language, 이하 XML이라 칭함) 기반 망 관리 시스템 및 방법에 관한 것으로, 상기 망 관리 시스템은, 네트워크 장비가 지원하는 CLI(Command Line Interface) 명령어의 형식을 XML(Extensible Markup Language)로 표현한 XML 템플릿; 및 상기 XML 템플릿을 트리 형태의 내부 자료구조로 변환한 후, 변환된 상기 XML 템플릿에게 소정의 인자를 전달하여, 상기 네트워크 장비에게 전송 가능한 CLI 명령어의 집합으로 변환하는 망 관리 인터페이스를 포함한다.

【대표도】

도 1

【명세서】**【발명의 명칭】**

이기종 네트워크 장비의 구성 관리를 위한 엑스엠엘 기반 망 관리 시스템 및 방법

{XML based network management system and method for configuration management of heterogeneous network devices}

【도면의 간단한 설명】

도 1은 본 발명의 바람직한 실시예에 따른 망 관리 시스템의 블록도이다.

도 2는 도 1에 도시된 망 관리 인터페이스의 상세 구조를 보여주는 도면이다.

도 3은 도 1에 도시된 XML 템플릿의 문법 구조를 보여주기 위한 도면이다.

도 4는 CLI 명령어의 일 예 및, 상기 CLI 명령어를 XML 템플릿으로 표현한 일 예를 보여주는 도면이다.

도 5는 본 발명에 따른 CLI 명령어의 물리화 방법을 설명하기 위한 도면이다.

도 6은 본 발명에 따른 망 관리 인터페이스 방법에 의해 물리화된 CLI 명령어들의 수행 과정을 보여주기 위한 흐름도이다.

도 7은 CLI 명령어의 물리화 및 물리화된 CLI 명령어들의 수행시 본 발명에 따른 망 관리 시스템의 각 구성 요소들이 호출되는 과정을 보여주기 위한 도면이다.

< 도면의 주요 부분에 대한 부호의 설명 >

100 : 망 관리 시스템 110 : XML 템플릿

120 : 망 관리 인터페이스 130 : 변환된 XML 템플릿

140 : 물리화된 CLI 명령어 160 : 네트워크 장비

170 : TELNET

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <13> 본 발명은 망 관리 시스템 및 방법에 관한 것으로, 특히 망 관리 시스템의 유지 보수성을 향상시킬 수 있는 엑스엠엘(Extensible Markup Language, 이하 XML이라 칭함) 기반 망 관리 시스템 및 방법에 관한 것이다.
- <14> 네트워크 장비에 구성 명령을 전송하기 위한 방법으로는 SNMP(Simple Network Management Protocol), COPS(Common Object Policy Service) 프로토콜 등과 같은 여러 가지 방법이 있다. SNMP는 비교적 간단하고 이해하기 쉬운 프로토콜이라는 장점이 있으나, 최근 들어 더욱 복잡해지고 있는 네트워크 관련 정책 정보(policy information)들을 네트워크 장비에 전달하기에는 지나치게 단순하고 보안관련 기능이 미흡한 단점이 있다. COPS 프로토콜은 이점을 극복하기 위해 IETF(Internet Engineering Task Force)에서 제안된 프로토콜로서, 정책 정보를 네트워크 장비에 전송하기 위해, 잘 정의된 형태의 객체지향적 메시지 구조를 가진다. 그러나, 아직도 많은 장비들은 COPS 프로토콜 대신 SNMP를 지원하고 있다.
- <15> 반면, 장비를 관리하기 위한 TELNET 기반의 수동 인터페이스인 CLI(Command Line Interface)는 네트워크 장비에 대한 관리를 위한 가장 기본적인 방법으로, 네트워크 장비에 대한 원격 관리를 위한 가장 일반적인 방법이다. 그러나, CLI는 사용자에게 명령어를 입력하는 방식으로, 사용자의 편의성이 떨어지며, 네트워크 장비에 대한 관리가 복잡한 경우 CLI를 사용하는 것은 매우 번거로울 수 있다. 특히, 네트워크 장비에 대한 관리가 복잡한 경우, CLI를 사용하는 것은 매우 번거로울 수 있다. 특히, 네트워크 장비에 대한 관리가 복잡한 경우, CLI를 사용하는 것은 매우 번거로울 수 있다.

Line Interface)는, 거의 모든 장비에서 지원하고 있고, 최신 표준을 가장 빠르게 반영하고 있다. 따라서, 대다수의 망 관리 시스템들은 장비를 관리하기 위해 TELNET 프로토콜로 장비에 접속 한 뒤, 정책 정보를 일련의 CLI 명령어들로 변환하여 장비에 전송하는 비 표준적 대안을 채택하고 있다. 그러나 이 같은 CLI 기반의 망 관리 시스템들은 정책 정보를 프로그램 상에서 CLI 명령어로 변환하는 방안을 채택하고 있기 때문에, 망 관리 소프트웨어의 프로그램이 CLI 문법에 직접적으로 종속되는 단점을 가지고 있다. 따라서, 장비의 OS(operating system)가 최신판으로 업그레이드되는 경우, 해당 CLI 문법의 변화를 반영하기 위해 망 관리 시스템을 변경해서 구현해야 하는 문제가 발생하게 된다. 그러므로, 이와 같은 단점을 극복하고, CLI 기반 망 관리 시스템의 유지 보수성을 향상시킬 수 있는 새로운 방안이 요구된다.

【발명이 이루고자 하는 기술적 과제】

- <16> 본 발명이 이루고자 하는 기술적 과제는, 메모리에 저장된 XML 템플릿과, XML 템플릿에 전달된 인자를 결합하여 물리화된 CLI 명령어들을 생성하여 망 관리 소프트웨어의 프로그램이 CLI 문법에 종속되지 않도록 함으로써, CLI 기반 망 관리 시스템의 유지 보수성을 향상시킬 수 있는 망관리 시스템 및 방법을 제공하는데 있다.
- <17> 본 발명이 이루고자 하는 다른 기술적 과제는, 상기 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하는데 있다.

【발명의 구성 및 작용】

- <18> 상기의 과제를 이루기 위하여 본 발명에 의한 망 관리 시스템은, 네트워크 장비가 지원하는 CLI(Command Line Interface) 명령어의 형식을 XML(Extensible Markup

Language)로 표현한 XML 템플릿; 및 상기 XML 템플릿을 트리 형태의 내부 자료구조로 변환한 후, 변환된 상기 XML 템플릿에게 소정의 인자를 전달하여, 상기 네트워크 장비에게 전송 가능한 CLI 명령어의 집합으로 변환하는 망 관리 인터페이스를 포함하는 것을 특징으로 한다.

<19> 상기의 과제를 이루기 위하여 본 발명에 의한 망 관리 방법은, (a) 네트워크 장비가 지원하는 CLI(Command Line Interface) 명령어의 형식을 XML(Extensible Markup Language)로 표현한 XML 템플릿을 구성하는 단계; 및 (b) 상기 XML 템플릿을 트리 형태의 내부 자료구조로 변환한 후, 변환된 상기 XML 템플릿에게 소정의 인자를 전달하여, 상기 네트워크 장비에게 전송 가능한 CLI 명령어의 집합으로 변환하는 단계를 포함하는 것을 특징으로 한다.

<20> 바람직한 실시예에 있어서, 상기 (b) 단계는 (b-1) 상기 XML 템플릿을 트리 형태의 상기 내부 자료 구조로 변환하는 단계; (b-2) 변환된 상기 XML 템플릿에 소정의 인자를 전달하여 상기 XML 템플릿을 상기 CLI 명령어의 집합으로 변환하는 단계; (b-3) 변환된 상기 CLI 명령어들을 상기 네트워크 장비에게 전송하는 단계; 및 (b-4) 전송된 상기 CLI 명령어들의 수행 성공/실패 여부 및 부가 정보를 수집하는 단계를 포함하는 것을 특징으로 한다.

<21> 이하에서, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 대하여 상세히 설명한다.

<22> 도 1은 본 발명의 바람직한 실시예에 따른 망 관리 시스템(100)의 블록도로서, 망 관리 인터페이스(120)를 이용하여 망 관리 시스템(100)을 구성하는 방법, 및 망 관리 시스템(100)과 네트워크와의 관계를 보여주고 있다.

<23> 도 1을 참조하면, 망 관리 시스템(100)은 X-CLI 인터페이스를 망 관리 인터페이스(120)로 사용하여 XML 템플릿(110)을 메모리에 적재한다. XML 템플릿(110)은 네트워크 장비가 지원하는 CLI 명령어의 형식을 XML로 표현한 것으로, 특정 기능을 장비에서 수행시키기 위해 필요한 CLI 명령어들의 연관관계가 기술되어 있다. 이 템플릿(110)은 트리 형태의 내부 자료구조(130)로 변환된 후, 망 관리 인터페이스(120)에서 내부적으로 관리된다. 망 관리 시스템(100)이 망 관리 인터페이스(120)를 이용하여 변환된 XML 템플릿(130)에게 해당 인자를 전달하게 되면, 상기 템플릿(130)은 네트워크 장비(160)에 전송 가능한 CLI 명령어의 집합으로 변환된다. 이를 물리화된 CLI 명령어들(140)이라고 부른다. 이와 같이, CLI 문법에 종속적인 부분을 XML 템플릿(110)이라는 파일(file) 형태로 분리한 뒤, 이를 실제 CLI 명령어들로 변환하는 과정을 자동화하게 되면, 망 관리 소프트웨어를 CLI 문법으로부터 비 종속적으로 만들 수 있게 된다.

<24> 물리화된 CLI 명령어들(140)은 망 관리 인터페이스(120)를 호출하여 네트워크 장비(160)에게 전송된다. 망 관리 시스템(100)과 네트워크 장비(160)는 망 관리 인터페이스(120)에 의해 설정된 TELNET 연결(170)을 통해서 통신을 수행하게 되며, 상기 통신을 수행하기 위한 프로토콜로는 TELNET 외에도 보안 쉘(secure shell ; SSH) 등과 같이 네트워크 장비(160)에 대해 가상 터미널(virtual terminal) 기능을 제공하는 프로토콜은 무엇이든 통신에 이용될 수 있다.

<25> 도 2는 도 1에 도시된 망 관리 인터페이스(120)의 상세 구조를 보여주는 도면이다. 도 2를 참조하면, 망 관리 인터페이스(120)는 XML 분석기(XML parser ; 121), 물리화기(materializer ; 122), 전송 처리기(connection manager ; 123), 및 처리결과 분석기(result processor ; 124)로 구성된다.

- <26> XML 분석기(121)는 XML 템플릿(110)을 메모리에 적재하고, 이를 트리 형태의 내부 자료 구조(130)로 변환하는 역할을 담당한다. 물리화기(122)는 변환된 XML 템플릿(130)에 인자를 전달하여 CLI 명령어들로 변환하는(즉, 물리화하는) 역할을 담당한다. 전송 처리기(123)는 물리화된 CLI 명령어들(140)을 네트워크 장비(160)에게 전송하는 역할을 담당한다. 그리고, 처리결과 분석기(124)는 전송된 CLI 명령어들(140)의 수행 결과를 분석하여, CLI 명령어(140)의 수행의 성공/실패 여부를 판정하거나, 수행 결과로부터 얻을 수 있는 부가적 정보를 수집하는 역할을 담당한다.
- <27> 도 3은 도 1에 도시된 XML 템플릿(110)의 문법 구조를 보여주기 위한 도면으로, 본 발명에 따른 XML 템플릿(110)은 XML 표준의 하나인 DTD(Document Type Declaration)를 이용하여 기술된다. DTD는 XML 문서를 구성하는 태그(tag)의 목록과, 해당 태그의 속성(attribute)들을 나열하기 위해 사용되며, 도 3에 도시된 DTD는 XML 템플릿(110)의 문법을 ELEMENT 태그, ATTLIST 태그, CDATA 태그, 및 #IMPLIED 태그로서 기술한다.
- <28> 먼저, ELEMENT 태그는, XML 문서에 <cli> 태그가 나타날 수 있다는 것과, 해당 <cli> 태그가 0개 이상의 하위 <cli> 태그를 포함하거나, 문자열 데이터를 포함할 수 있다는 것을 나타낸다.
- <29> ATTLIST 태그는 <cli> 태그의 속성을 명시하는 데 사용된다. ATTLIST 태그에 의해 명시되는 <cli> 태그의 속성으로는, tag, prompt1, prompt2, command, errorstr, always, ainput, 및 ainputresponse 등이 있다.
- <30> CDATA는 해당 속성이 문자열 값을 가짐을 나타내고, #IMPLIED로 명시된 속성들은 생략될 수 있는 것을 나타낸다. 도 3에서, always 태그의 경우처럼 #IMPLIED 대신 어떤 값이 명시된 경우, 해당 값이 내정 값(default value)으로서 사용된다. 즉, <cli> 태그

이용시 해당 속성의 값을 적어주지 않으면, 해당 내정 값이 주어진 것으로 가정된다. <cli> 태그가 가지는 각각의 속성들을 개별적으로 살펴보면 다음과 같다.

<31> tag는 문서 내에서 해당 <cli> 태그를 식별해 내기 위한 식별자로 사용된다. 이는 여러 개의 <cli> 태그가 동시에 존재할 때, 인자 전달을 위해 특정 태그를 식별하는 데 사용된다. command는 실제 CLI 명령어를 나타내기 위한 속성이며, prompt1은 command 속성 값으로 명시된 CLI 명령어를 전송하기 위해 해당 프롬프트 문자열을 네트워크 장비(160)의 가상 터미널(170) 상에 나타내는 데 사용되는 속성이다. prompt2는 CLI 명령어를 전송하여 수행한 뒤 어떤 프롬프트가 네트워크 장비(160)의 가상 터미널(170) 상에 나타나야 하는지를 명시하는 데 사용된다. errorstr은 CLI 명령어를 수행한 결과가 실패일 경우, 네트워크 장비(160)의 가상 터미널(170) 상에 어떤 문자열이 나타날 수 있는지를 명시하는 데 사용되며, 이 속성 값을 이용하여 개별적 CLI 명령어의 수행에 대한 성공/실패 여부를 판정하게 된다. always는 앞서 실행된 임의의 CLI 명령어가 실패하였을 경우, 현 CLI 명령어를 수행할 것인지 아닌지의 여부를 결정하는 데 사용된다. 예를 들어, always 속성의 값이 true인 경우, 해당 CLI 명령어는 이전에 수행된 CLI 명령어의 수행 결과가 성공이든 실패이든 상관없이 수행된다. 그리고, false인 경우에는, 이전에 수행된 CLI 명령어들이 모두 성공적으로 수행된 경우에만 수행된다. 이 같은 always 속성은 기본 값으로 false 값을 가진다.

<32> 이 외에도, 상기 속성들이 모두 생략된 <cli> 태그를 순수 그룹화 태그(pure aggregation tag)라 부르며, 약어로 PAT라고 한다. 순수 그룹화 태그는, 해당 태그에 의해 포함된 하위 <cli> 태그들이 복수 회 몰리화될 수 있음을 나타낸다. 다시 말해, 인자

들이 계속해서 주어지게 되면, 해당 태그들은 여러 번 물리화될 수 있다는 것을 의미하게 된다.

<33> 도 4는 CLI 명령어의 일 예(112) 및, 상기 CLI 명령어를 XML 템플릿(110)으로 표현한 일 예(113)를 보여주는 도면이다. 도 4를 참조하면, 도 4에 도시된 CLI 명령어(112)는 Cisco Systems, Inc.의 PE(Provider Edge) 라우터에 BGP(Border Gateway Protocol) 관련 기능을 설정하는 명령어의 일 예이다. 이 예는, CLI 명령어들이 어떻게 계층적으로 연관되는지를 잘 보여준다.

<34> 예를 들어, address-family ipv4 vrf VRF-SEOUL과 같은 명령어는, 상위 명령어인 router bgp 55555가 수행되지 않으면 입력될 수 없다. 이와 같은 방식의 계층적 연관 관계를 계층적 종속성(hierarchical dependency)이라 한다. 또한, 어떤 명령어가 수행되다 실패할 경우, 그 이후의 CLI 명령어들은 입력할 수 없기 때문에, 후행 명령어의 실행 여부가 선행 명령어의 수행 결과에 종속되는 현상이 발생한다. 이런 연관 관계를 결과 종속성(result dependency)이라 한다. 아울러, router bgp 55555 같은 명령어에서 55555 같은 부분은 router bgp 명령어에 대한 인자(argument)인 것으로 생각할 수 있는데, 이 인자가 생략 될 경우 router bgp 명령어는 실행될 수 없으므로 인자와 명령어의 수행 여부간에 종속성이 발생하게 된다. 이런 종속성을 인자 종속성(argument dependency)이라 한다.

<35> 따라서, 본 발명에서는 도 3에서 제시된 XML 템플릿(110)의 문법을 이용하여 이와 같은 CLI 명령어들이 가지고 있는 계층적 종속성과, 인자 종속성 및 결과 종속성에 관한 내용을 XML 형식으로 기술한다. 도 3에서 제시된 XML 템플릿(110)의 문법을 근거로 하여

, 도 4에 도시된 CLI 명령어(112)를 XML 템플릿으로 표현하게 되면 참조번호 113과 같이 기술된다.

<36> CLI 명령어의 계층적 종속성은 <cli> 태그간 포함 관계에 의해 기술된다. 예를 들어, 임의의 CLI 명령어에 포함된 CLI 명령어는, 자신을 포함하는 CLI 명령어가 물리화되지 않으면 물리화될 수 없고, 자신을 포함하는 CLI 명령어가 성공적으로 수행되지 않으면, 물리화 되더라도 수행될 수 없게 된다. 이런 점에 비추어 볼 때, 물리화는 일반 프로그래밍 언어에서의 컴파일(compile)과 동등한 개념이며, 그 수행 역시 일반 프로그래밍 언어에서의 프로그램 수행(execution)과 동등한 개념이라 할 수 있다. 물리화 과정에 대해서는 도 5를 참조하여 아래에서 상세히 설명될 것이다.

<37> CLI 명령어의 인자 종속성은 command 속성 값에 포함된 \$로 시작하는 형식 인자명에 의해 표현된다. 이 형식 인자명은 추후, X-CLI API(Application Programming Interface)에 의해서 XML 템플릿(110)이 메모리에 적재된 후, 적재된 XML 템플릿(130)에 전달된 인자에 의해 대치된다. 인자가 적절하게 전달되지 않은 경우, 해당 CLI 명령어는 물리화 될 수 없게 된다.

<38> CLI 명령어의 결과 종속성은 <cli> 태그의 errorstr 속성과 always 속성에 의해 표현된다. 앞서 수행된 물리화된 CLI 명령어의 수행이 실패했을 경우(예를 들어, 수행 결과에 errorstr 속성에 명시된 값이 포함된 경우, 실패한 것으로 간주된다), 이후 수행되는 CLI 명령어들은, always 속성 값이 true인 것으로 제한된다. 결과 종속성에 관한 보다 자세한 사항은 도 6을 참조하여 아래에서 설명될 것이다.

<39> 도 5는 본 발명에 따른 CLI 명령어의 물리화 방법을 설명하기 위한 도면이다. 본 발명에 따른 CLI 명령어의 물리화 과정은, 도 2에 도시된 망 관리 인터페이스(120)의 물

리화기(122)에 의해서, 메모리에 트리 형태의 내부 데이터 구조로 저장되어 있는 XML 템플릿(130)과, 템플릿(130)에 주어진 인자를 결합함에 의해, 물리화된 CLI 명령어 시퀀스(140)를 생성하게 된다. 물리화된 CLI 명령어 시퀀스(140)는 다음과 같은 과정에 의해 얻어지게 된다.

- <40> 먼저, 물리화를 수행해야 할 대상인 XML 템플릿이 CLI 명령어들의 트리 형태로 주어지고(141), 인자들의 큐(142)가 주어진다. 만일 물리화를 시작할 트리의 뿌리 노드(root node)에 해당하는 <cli> 태그가 PAT가 아니면, 인자가 필요한 <cli> 태그인지 여부를 검사한다. 검사 결과, 인자가 필요한 태그이면, 인자들의 큐(142)를 뒤져 필요한 인자가 큐(142) 내에 존재하는지를 검사한다. 검사 결과, 인자가 큐(142) 내에 존재하는 경우에 는 해당 인자를 <cli> 태그에 적용하여 물리화한다. 이와 같은 과정을 통해 뿌리 노드에 대한 물리화가 종료되면, 뿌리 노드의 모든 자식 노드(child node)들에게 이와 같은 물리화 과정을 재귀적으로 적용하게 된다.
- <41> 물리화 과정의 적용이 성공적으로 끝나고 난 뒤, 뿌리 노드가 PAT로 판명되면, 자식 노드들이 다시 물리화될 가능성이 있으므로 restart:로 표시된 위치부터 상기 과정을 다시 수행한다. 이와 같은 물리화 과정을 도 4의 153에 적용한 결과는 도 5의 143과 같다.
- <42> 앞에서 설명한 바와 같이, 본 발명에 따른 CLI 명령어의 물리화 방법은, XML 템플릿(110)에 의해 구성되는 CLI 명령어들간에 존재하는 트리 형태의 계층적 연관성을 연속적인 CLI 명령어 실행 궤적으로 변환한다. 그러나, 앞에서 설명한 바와 같이 어떤 CLI 명령어가 수행 도중 실패하게 되면, 계층적으로 하위에 속하는 명령어들은 수행되지 않아야 한다. 따라서, 본 발명에서는 실패시의 분기 위치가 각각의 CLI 명령어를 위한 추

가 속성으로 자동 생성되어 관리된다. 이 같은 실패시의 분기 위치는 공통의 부모 노드를 공유하는 다음번 형제 노드(sibling node)가 된다. 도 5에 도시된 물리화된 CLI 명령어 시퀀스(140)는 해당 분기 위치들이 어떻게 생성되었는지를 잘 보여준다. 물리화된 CLI 명령어(140)의 수행 순서는 다음과 같다.

<43> 도 6은 본 발명에 따른 망 관리 인터페이스 방법에 의해 물리화된 CLI 명령어들의 수행 과정을 보여주기 위한 흐름도로서, 도 2에 도시된 전송 처리기(123)에 의해 물리화된 실제 CLI 명령어들(140)이 네트워크 장비(160)에 하나씩 전송되어 수행되는 과정이 도시되어 있다.

<44> 도 6을 참조하면, 전송 처리기(123)는 먼저 failed 변수 값을 false로 만들어 두고, 변수 i 는 물리화된 실제 CLI 명령어 중 첫 번째 명령어의 주소 값을 갖도록 지정한다 (1201 단계). 그리고 나서 i가 0인지 여부를 판별한다(1202 단계).

<45> 1202 단계에서의 판별 결과, i가 0이면 CLI 명령어들의 수행을 종료하고, i가 0이 아니면 i가 0이 아닌 동안(즉, i가 유효한 명령어를 가리키는 동안) 루프를 돌면서 아래의 과정을 수행한다. 이를 위해, 먼저 1201 단계에서 설정된 failed 변수 값이 false를 그대로 유지하고 있는지 여부가 판별된다(1203 단계).

<46> 1203 단계에서의 판별 결과, 1201 단계에서 설정된 failed 변수 값이 false를 그대로 유지하고 있으면, 네트워크 장비(160)로부터 prompt1 속성 값으로 명시된 프롬프트 문자열이 전송될 때까지 대기한다(1204 단계). 이어서, 해당 프롬프트 문자열이 전송되면 CLI 명령어(140)를 전송하고, 해당 CLI 명령어(140)의 수행 결과를 수집한다(1205 단계). 그리고 나서, 네트워크 장비(160)가 추가 입력을 요구하는지 여부(즉, ainput 속성 값이 네트워크 장비(160)로부터 전송되었는지 여부)를 판별한다(1206 단계).

- <47> 1206 단계에서의 판별 결과, 네트워크 장비(160)가 추가 입력을 요구하는 경우, 네트워크 장비(160)에게 추가 입력으로서 ainputresponse 속성 값으로 명시된 문자열을 전송한다(1207 단계). 그리고, 1206 단계에서의 판별 결과, 네트워크 장비(160)가 추가 입력을 요구하지 않는 경우, CLI 명령어의 수행에 오류가 발생되었는지 여부(즉, 네트워크 장비(160)로부터 errorstr 속성 값이 전송되었는지 여부)가 판별된다(1208 단계).
- <48> 1208 단계에서의 판별 결과, CLI 명령어의 수행에 오류가 발생되었으면, failed 변수 값을 true로 변경한다. 그리고, 1208 단계에서의 판별 결과, CLI 명령어의 수행에 오류가 발생되지 않았으면, failed 변수의 상태와 CLI 명령어(140)의 수행 실패시의 분기 위치(BTEF)를 고려한다(1210 단계).
- <49> 1210 단계에서 수행된 failed 변수의 상태와 CLI 명령어(140)의 수행 실패시 분기 위치(BTEF)의 고려 결과, failed 변수 값이 true이고, CLI 명령어(140)의 수행 실패시의 분기 위치(BTEF)가 0의 값을 가지는 경우, CLI 명령어(140)의 수행 실패시 분기 위치(BTEF)를 i에 저장하고, 1202 단계로 되돌아간다(1211 단계). 그리고, 1210 단계에서의 고려 결과, failed 변수 값이 true이고, CLI 명령어(140)의 수행 실패시의 분기 위치(BTEF)가 0의 값을 가지지 않는 경우, 다음 번 수행될 CLI 명령어의 주소값(NEXT)을 i에 저장하고, 1202 단계로 되돌아간다(1211 단계).
- <50> 도 7은 CLI 명령어의 물리화 및 물리화된 CLI 명령어들의 수행시 본 발명에 따른 망 관리 시스템(100)의 각 구성 요소들이 호출되는 과정을 보여주기 위한 도면이다.
- <51> 도 7을 참조하면, 본 발명에 따른 망 관리 시스템(100)에서 수행되는 동작은, 망 관리 인터페이스(120)를 통해 XML 템플릿(110)을 로드하기 위한 요청을 수행하는 과정,

XML 템플릿(110)에 대해 인자를 전달하는 과정, 및 물리화된 CLI 명령어(140)를 전송하는 과정과 같은 세 가지 단계로 정리할 수 있다.

<52> XML 템플릿(110)을 CLI 명령어(140)로 물리화 하는 과정은 실제 전송이 이루어지기 전, 망 관리 인터페이스(120)에 구비된 물리화기(122)에 의해 내부적으로 이루어진다.

이와 같은 망 관리 시스템(100)의 각 구성 요소들이 CLI 명령어의 물리화를 수행하고, 물리화된 CLI 명령어들을 수행할 때 어떤 절차로 호출되는지를 살펴보면 다음과 같다.

<53> 먼저, 망 관리 시스템(100)으로부터 XML 템플릿(110)에 대해 적재 요구가 발생되면, 상기 적재 요구는 망 관리 인터페이스(120)를 통해 XML 분석기(121)로 전달된다. XML 분석기(121)는 XML 템플릿(110)을 트리 형태의 내부 자료 구조(130)로 변환하여 메모리에 적재하고, XML 템플릿(110)이 메모리에 적재되었음을 망 관리 인터페이스(120)를 통해 망 관리 시스템(100)에게 전달한다.

<54> 망 관리 시스템(100)은 망 관리 인터페이스(120)를 통해 XML 템플릿에게 인자를 전달하고, 망 관리 인터페이스(120)는 인자를 큐에 저장하여 물리화기(122)에게 전달한다. 물리화기(122)는 변환된 XML 템플릿(130)에 인자를 전달하여 CLI 명령어들로 변환하는(즉, 물리화하는) 역할을 수행한다. 이 때, 망 관리 시스템(100)이 망 관리 인터페이스(120)를 통해 물리화된 CLI 명령어들에 대한 전송을 요구하게 되면, 물리화기(122)는 물리화가 완료된 후 전송처리기(123)에게 전송을 요구하게 된다.

<55> 전송 처리기(123)는 물리화기(122)로부터 입력된 전송 요구에 응답해서, 물리화된 CLI 명령어들(140)을 네트워크 장비(160)에게 전송한다. 네트워크 장비(160)는 CLI 명령어들(140)에 대한 전송 결과를 처리결과 분석기(124)에게 전송하고, 처리결과 분석기(124)는 상기 전송 결과에 응답해서 CLI 명령어들(140)의 수행이 성공했는지 실패했는

지를 분석한다. 그리고, 분석결과를 전송처리기(123)에게 전달한다. 전송처리기(123)는 처리결과 분석기(124)로부터 입력된 분석결과에 응답해서 CLI 명령어들(140)의 수행 결과를 망 관리 인터페이스(120)에게 전달하고, 오류 발생시에는 오류 내용을 보고하는 역할을 수행한다.

<56> 이와 같은 본 발명에 따른 일련의 망 관리 동작에 의하면, 프로그램 상에서 정책 정보를 CLI 명령어로 변환하는 대신, CLI 명령어가 가지고 있는 계층적 구조를 잘 나타낼 수 있는 XML 템플릿을 구성하여 이를 실제 CLI 명령어로 생성함으로써, 망 관리 소프트웨어가 CLI 문법에 종속되지 않도록 한다. 따라서, 장비의 OS가 업그레이드되더라도 망 관리 시스템을 변경할 필요가 없을 뿐만 아니라, 망 관리 시스템의 유지 보수성을 증가시켜 줄 수 있다. 그러므로, SNMP(Simple Network Management Protocol), COPS(Common Object Policy Service) 프로토콜 등을 사용하는 이기종 네트워크 장비들에 대해서도 CLI 기반의 망 관리 시스템을 용이하게 구현할 수 있고, 좀 더 손쉽게 관리할 수 있게 된다.

<57> 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드로 저장되고 실행될 수 있다.

【발명의 효과】

<58> 이상에 설명한 바와 같이, 본 발명에 의한 엑스엠엘 기반 망 관리 시스템 및 방법에 의하면, 메모리에 저장된 XML 템플릿과, XML 템플릿에 전달된 인자를 결합하여 물리화된 CLI(Command Line Interface) 명령어를 생성하고, 이를 네트워크 장비에게 정책 정보를 전달하는데 사용함으로써, 망 관리 시스템의 유지 보수성을 증가시켜 준다. 따라서, 망 관리 소프트웨어를 관리할 때 발생될 수 있는 유지보수 문제 및 오류 문제를 줄임으로써, 네트워크 장비를 좀 더 손쉽게 관리할 수 있다

【특허청구범위】**【청구항 1】**

네트워크 장비가 지원하는 CLI(Command Line Interface) 명령어의 형식을 XML(Extensible Markup Language)로 표현한 XML 템플릿; 및 상기 XML 템플릿을 트리 형태의 내부 자료구조로 변환한 후, 변환된 상기 XML 템플릿에게 소정의 인자를 전달하여, 상기 네트워크 장비에게 전송 가능한 CLI 명령어의 집합으로 변환하는 망 관리 인터페이스를 포함하는 것을 특징으로 하는 망 관리 시스템.

【청구항 2】

제 1 항에 있어서, 상기 망 관리 인터페이스는 상기 XML 템플릿을 트리 형태의 상기 내부 자료 구조로 변환하는 XML 분석기; 변환된 상기 XML 템플릿에 소정의 인자를 전달하여 상기 XML 템플릿을 상기 CLI 명령어의 집합으로 변환하는 물리화기; 변환된 상기 CLI 명령어들을 상기 네트워크 장비에게 전송하는 전송 처리기; 및 전송된 상기 CLI 명령어들의 수행 성공/실패 여부 및 부가 정보를 수집하는 처리결과 분석기를 포함하는 것을 특징으로 하는 망 관리 시스템.

【청구항 3】

제 1 항에 있어서,
상기 망 관리 인터페이스는 X-CLI 인터페이스인 것을 특징으로 하는 망 관리 시스템.

【청구항 4】

제 1 항에 있어서,

상기 망 관리 인터페이스와 상기 네트워크 장비는, 상기 네트워크 장비에 대해 가상 터미널(virtual terminal) 기능을 제공하는 프로토콜을 통해 연결되는 것을 특징으로 하는 망 관리 시스템.

【청구항 5】

제 1 항에 있어서,

상기 XML 템플릿은, XML 문서를 구성하는 태그의 목록과, 해당 태그의 속성들을 나열하는 데 사용되는 DTD(Document Type Declaration)를 이용하여 기술되는 것을 특징으로 하는 망 관리 시스템.

【청구항 6】

제 5 항에 있어서, 상기 XML 템플릿은,

상기 XML 문서에 CLI 태그가 나타날 수 있다는 것과, 해당 CLI 태그가 하위 CLI 태그 또는 문자열 데이터를 포함할 수 있다는 것을 나타내기 위한 제 1 태그;

상기 CLI 태그의 속성을 명시하기 위한 제 2 태그;

상기 제 2 태그에 의해 명시되는 해당 속성이 문자열 데이터를 가짐을 나타내는 제 3 태그; 및

상기 제 2 태그에 의해 명시되는 해당 속성이 생략될 수 있음을 나타내는 제 4 태그를 포함하는 것을 특징으로 하는 망 관리 시스템.

【청구항 7】

제 6 항에 있어서, 상기 제 2 태그에 의해 명시되는 상기 CLI 태그의 속성은, 복수 개의 CLI 태그가 동시에 존재할 때 특정 태그를 식별하기 위한 제 1 속성; 상기 CLI 명령어를 실제적으로 나타내기 위한 제 2 속성; 상기 제 2 속성 값으로 명시된 상기 CLI 명령어를 전송하기 위해 해당 프롬프트 문자열을 상기 네트워크 장비의 가상 터미널 상에 나타내기 위한 제 3 속성; 상기 CLI 명령어를 전송하여 수행한 뒤 상기 네트워크 장비의 상기 가상 터미널 상에 어떤 프롬프트가 표시되어야 하는지를 나타내는 제 4 속성; 상기 CLI 명령어의 수행 결과가 실패일 경우 상기 네트워크 장비의 상기 가상 터미널 상에 어떤 문자열이 표시되는지를 명시하여, 상기 CLI 명령어의 수행에 대한 성공/실패 여부를 판정하는 제 5 속성; 및 이전 CLI 명령어의 수행 실패시 현 CLI 명령어를 수행할 것인지 여부를 결정하기 위한 제 6 속성을 포함하는 것을 특징으로 하는 망 관리 시스템.

【청구항 8】

제 7 항에 있어서,

상기 CLI 태그의 속성들이 모두 생략된 CLI 태그는, 상기 태그에 포함된 하위 CLI 태그들이 복수 회 물리화될 수 있는 순수 그룹화 태그(pure aggregation tag)인 것을 특징으로 하는 망 관리 시스템.

【청구항 9】

- (a) 네트워크 장비가 지원하는 CLI(Command Line Interface) 명령어의 형식을 XML(Extensible Markup Language)로 표현한 XML 템플릿을 구성하는 단계; 및
- (b) 상기 XML 템플릿을 트리 형태의 내부 자료구조로 변환한 후, 변환된 상기 XML 템플릿에게 소정의 인자를 전달하여, 상기 네트워크 장비에게 전송 가능한 CLI 명령어의 집합으로 변환하는 단계를 포함하는 것을 특징으로 하는 망 관리 방법.

【청구항 10】

제 9 항에 있어서, 상기 (b) 단계는

- (b-1) 상기 XML 템플릿을 트리 형태의 상기 내부 자료 구조로 변환하는 단계;
- (b-2) 변환된 상기 XML 템플릿에 소정의 인자를 전달하여 상기 XML 템플릿을 상기 CLI 명령어의 집합으로 변환하는 단계;
- (b-3) 변환된 상기 CLI 명령어들을 상기 네트워크 장비에게 전송하는 단계; 및
- (b-4) 전송된 상기 CLI 명령어들의 수행 성공/실패 여부 및 부가 정보를 수집하는 단계를 포함하는 것을 특징으로 하는 망 관리 방법.

【청구항 11】

제 9 항에 있어서,

상기 망 관리 인터페이스는 X-CLI 인터페이스인 것을 특징으로 하는 망 관리 방법.

【청구항 12】

제 9 항에 있어서,

상기 망 관리 인터페이스와 상기 네트워크 장비는, 상기 네트워크 장비에 대해 가상 터미널(virtual terminal) 기능을 제공하는 프로토콜을 통해 연결되는 것을 특징으로 하는 망 관리 방법.

【청구항 13】

제 9 항에 있어서,

상기 XML 템플릿은, XML 문서를 구성하는 태그의 목록과, 해당 태그의 속성들을 나열하는 데 사용되는 DTD(Document Type Declaration)를 이용하여 기술되는 것을 특징으로 하는 망 관리 방법.

【청구항 14】

제 13 항에 있어서, 상기 XML 템플릿은,

상기 XML 문서에 CLI 태그가 나타날 수 있다는 것과, 해당 CLI 태그가 하위 CLI 태그 또는 문자열 데이터를 포함할 수 있다는 것을 나타내기 위한 제 1 태그;

상기 CLI 태그의 속성을 명시하기 위한 제 2 태그;

상기 제 2 태그에 의해 명시되는 해당 속성이 문자열 데이터를 가짐을 나타내는 제 3 태그; 및

상기 제 2 태그에 의해 명시되는 해당 속성이 생략될 수 있음을 나타내는 제 4 태그를 포함하는 것을 특징으로 하는 망 관리 방법.

【청구항 15】

제 14 항에 있어서, 상기 제 2 태그에 의해 명시되는 상기 CLI 태그의 속성은,

복수 개의 CLI 태그가 동시에 존재할 때 특정 태그를 식별하기 위한 제 1 속성;

상기 CLI 명령어를 실제적으로 나타내기 위한 제 2 속성;

상기 제 2 속성 값으로 명시된 상기 CLI 명령어를 전송하기 위해 해당 프롬프트 문자열을 상기 네트워크 장비의 가상 터미널 상에 나타내기 위한 제 3 속성;

상기 CLI 명령어를 전송하여 수행한 뒤 상기 네트워크 장비의 상기 가상 터미널 상에 어떤 프롬프트가 표시되어야 하는지를 나타내는 제 4 속성;

상기 CLI 명령어의 수행 결과가 실패일 경우 상기 네트워크 장비의 상기 가상 터미널 상에 어떤 문자열이 표시되는지를 명시하여, 상기 CLI 명령어의 수행에 대한 성공/실패 여부를 판정하는 제 5 속성; 및

이전 CLI 명령어의 수행 실패시 현 CLI 명령어를 수행할 것인지 여부를 결정하기 위한 제 6 속성을 포함하는 것을 특징으로 하는 망 관리 방법.

【청구항 16】

제 15 항에 있어서,

상기 CLI 태그의 속성들이 모두 생략된 CLI 태그는, 상기 태그에 포함된 하위 CLI 태그들이 복수 회 물리화될 수 있는 순수 그룹화 태그(pure aggregation tag)인 것을 특징으로 하는 망 관리 방법.

【청구항 17】

제 10 항 또는 제 15 항에 있어서, 상기 (b-3) 단계는,

(b-3-1) 상기 CLI 명령어의 수행 실패를 나타내는 변수 값을 거짓으로 설정하고, 변수 i를 물리화된 첫 번째 CLI 명령어의 주소 값으로 설정하는 단계;

(b-3-2) 상기 변수 i 가 유효한 명령어를 가리키는 동안 상기 네트워크 장비로부터 상기 제 3 속성 값으로 명시된 소정의 프롬프트 문자열이 전송될 때까지 대기하는 단계;

(b-3-3) 상기 프롬프트 문자열이 전송되면 상기 CLI 명령어를 상기 네트워크 장비에게 전송하는 단계; 및

(b-3-4) 상기 네트워크 장비가 추가 입력을 요구하는 경우, 소정의 추가 문자열을 전송하는 단계를 포함하는 것을 특징으로 하는 망 관리 방법.

【청구항 18】

제 17 항에 있어서, 상기 (b-4) 단계는

(b-4-1) 상기 CLI 명령의 수행 결과, 오류가 발생한 경우 상기 CLI 명령어의 수행 실패를 나타내는 상기 변수 값을 참으로 설정하는 단계; 및

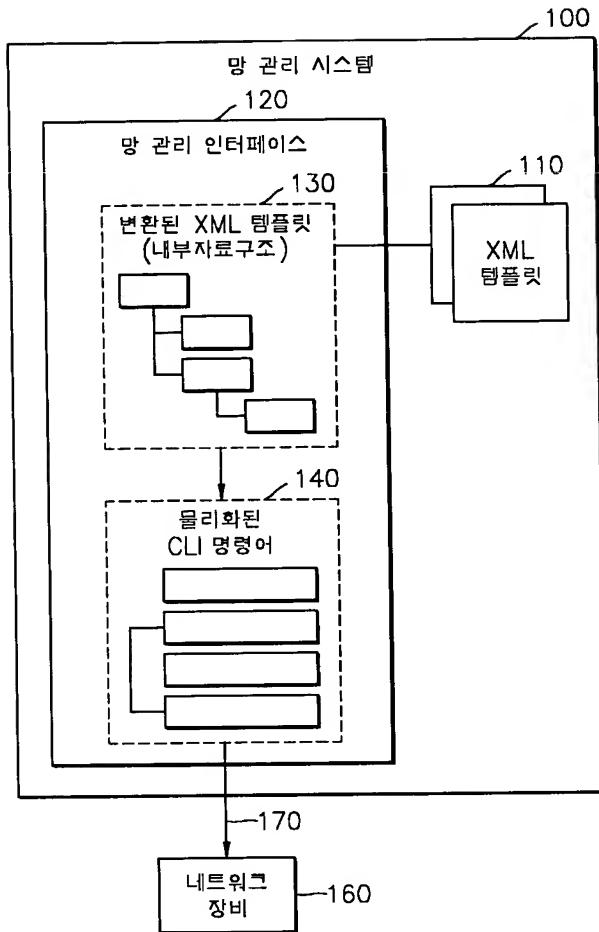
(b-4-2) 상기 CLI 명령어의 수행 실패를 나타내는 상기 변수 값의 상태 및 상기 CLI 명령어의 수행 실패시의 분기 위치를 참고하여, 다음 번 수행될 CLI 명령어의 주소 값을 상기 변수 i에 저장하는 단계를 포함하는 것을 특징으로 하는 망 관리 방법.

【청구항 19】

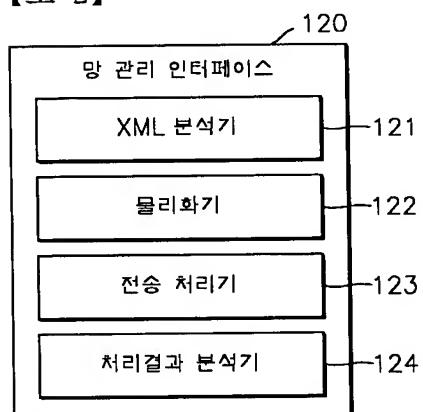
제 9 항 내지 제 18 항 중 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

【도면】

【도 1】



【도 2】



【도 3】

111

```
<!ELEMENT cli (#PCDATA | (cli)*)>
<!ATTLIST cli
tag          CDATA          #IMPLIED
prompt1      CDATA          #IMPLIED
prompt2      CDATA          #IMPLIED
command      CDATA          #IMPLIED
errorstr     CDATA          #IMPLIED
always       (true | false) "false"
ainput       CDATA          #IMPLIED
ainputresponse CDATA        #IMPLIED>
```

【도 4】

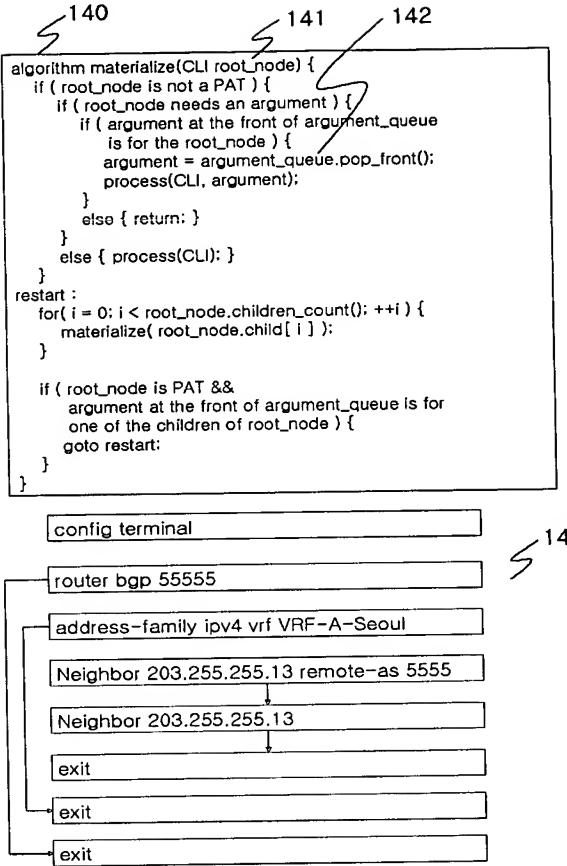
112

```
(config)# router bgp 55555
(config-router)# address-family ipv4 vrf VRF-SEOUL
(config-router-af)# neighbor 203.255.25.15 remote-as 5555
(config-router-af)# neighbor 203.255.25.15 activate
```

113

```
<cli prompt1="#" command="config terminal" errorstr="^">
<cli tag="bgp1" prompt1="#"
  command="router bgp $asnum" errorstr="^">
<cli>
<cli tag="bgp2" prompt1="#"
  command="address-family ipv4 vrf $vrfname"
  errorstr="^">
<cli>
<cli tag="bgp3" prompt1="#"
  command="neighbor $ipn1 remote-as $asnum"
  errorstr="^">
</cli>
<cli tag="bgp4" prompt1="#"
  command="neighbor $ipn2 activate"
  errorstr="^">
</cli>
</cli>
<cli prompt1="#" always="true" command="exit"></cli>
</cli>
<cli prompt1="#" always="true" command="exit"></cli>
</cli>
<cli prompt1="#" always="true" command="exit"></cli>
</cli>
```

【도 5】



【도 6】

